# APPLICATION FOR
# UNITED STATES PATENT
# IN THE NAME OF


Inventors

Qiang Shen

OF


# VIA TELECOM, INC.


FOR


# Single Engine Turbo Decoder with Single Frame Size Buffer for Interleaving/Deinterleaving


## DOCKET NO. VIA-007-CIP
## VIA DOCKET NO. VTB03-0002-US
(LSI Logic Docket No.: 02-0020)


*Prepared by*

Merle Richman and Marty Jaquez
--Express Mail No. ER 622956783 US--

# Single Engine Turbo Decoder with Single Frame Size Buffer for Interleaving/Deinterleaving

## CROSS REFERENCE TO RELATED APPLICATIONS

**001** This patent application is a continuation-in-part of Utility Patent Application 09/668,059, filed September 20, 2000, Attorney Docket Number 30454-00281, and entitled "Turbo Decoding", pending. This patent application claims the benefit under 35 USC §120 to the filing date of utility patent application 09/668,059.

## BACKGROUND

*1.    Field of the Invention*

**002** The invention relates generally to forward error correction and de-interleaver methods and apparatus in mobile communication networks, and more particularly, to a memory usage optimization of forward error correction and de-interleaver method and apparatus in mobile cellular communication networks.

*2.    Description of Related Art*

**003** In mobile cellular communication networks a forward error correction ("FEC") component is commonly determined and added to signals communicated between base stations and mobile units in the networks. In addition, the signal is commonly interleaved prior to transmission to reduce the effect of possible burst errors in the network. A mobile unit or base station receiving an interleaved signal with added FEC components must decode and de-interleave the signal. When concatenated code (turbo code) is used for FEC, the mobile unit employs a turbo decoder to process the code where the turbo decoder internally performs interleaving and de-interleaving. The turbo decoder commonly requires a large, energy inefficient memory. These inefficient memories increase integrated circuit (IC) real estate (*i.e.*, chip area) and power consumption of the mobile unit.

**004** A need thus exists for a mobile unit based FEC decoder and de-interleaver system and method that requires small power-efficient memory. The present invention provides such a mobile unit based FEC decoder and de-interleaver system and method.

## SUMMARY OF THE INVENTION

**005** The present invention includes an apparatus for and a method of decoding and deinterleaving a received signal, the received signal encoded with two constituent codes and interleaved on a frame by frame basis. The apparatus may include a single constituent code decoder and a single common buffer coupled to the single constituent code decoder where the common buffer sized to hold a single frame of received data. The apparatus may further include an address controller coupled to the single common buffer where the address controller generates read addresses that causes data to be de-interleaved when read from the common buffer. In one embodiment the address controller may also generate write addresses that cause data to be interleaved when written to the common buffer. In another embodiment, the common buffer may be divided into a plurality of sub-buffers where each sub-buffer is a single port memory.

**006** In an embodiment, the address controller may be configured to generate a read address for one of the plurality of sub-buffers and a write address for another of the plurality of sub-buffers, where the reading and writing of the respective sub-buffers occurs during the same clock cycle. The common buffer may also be divided into at least three sub-buffers where each sub-buffer is a single port memory. In this embodiment, the address controller may be configured to generate a read address for one of the sub-buffers and a write address for another of the plurality of sub-buffers so that the reading and writing of the respective sub-buffers occurs during the same clock cycle, and so that the remaining sub-buffers are idle during the same clock cycle. The address controller may be configured to generate read addresses using a one of a row-by-row with column shuffling algorithm and a column-by-column with row shuffling algorithm.

**007** A method of the present invention may serially decode the received signal and store received data and decoded data in a single common buffer where the common buffer is sized to hold a single frame of received data. The method may also generate read addresses that cause data to be de-interleaved when read from the common buffer. In another method, the read and write addresses can be generated such that the data interleaving or de-interleaving can be achieved by the combination of writing to the common buffer and reading from that buffer.

## BRIEF DESCRIPTION OF THE DRAWINGS

**008** The features, objects, and advantages of the present invention will become more apparent from the detailed description set forth below when taken in conjunction with the drawings in which like reference characters identify correspondingly throughout and wherein:

**009** FIGURE 1 is a block diagram of a cellular communication system comprising a plurality of base stations that may be used to practice the present invention.

**010** FIGURE 2 is a block diagram of an exemplary base station that may be used to practice the present invention.

**011** FIGURE 3 is a block diagram of an exemplary mobile unit that may be used to practice the present invention.

**012** FIGURE 4 is a simplified block diagram of an exemplary base station error encoder and interleaver that may be used to practice the present invention.

**013** FIGURE 5A is a block diagram of an exemplary mobile unit transceiver ASIC segment that may be used to practice the present invention.

**014** FIGURE 5B is a block diagram of a mobile unit de-interleaver and error decoder that may be employed in the transceiver ASIC shown in FIGURE 5A.

**015** FIGURE 6A is a block diagram of an exemplary mobile unit de-interleaver and error decoder that may be employed in the transceiver ASIC shown in FIGURE 5A.

**016** FIGURE 6B depicts an exemplary interleaving process that may be used to interleave data when stored in a memory in accordance with the teachings of the present invention.

**017** FIGURE 6C depicts an exemplary de-interleaving process that may be used to deinterleave data when stored in a memory in accordance with the teachings of the present invention.

**018** FIGURE 7 depicts an exemplary process that may be used to de-interleave and decode a received interleaved, error encoded signal in accordance with the teachings of the present invention.

**019** FIGURE 8 is a diagram showing an exemplary memory mapping/layout for use by the apparatus shown in FIGURE 6A and the process shown in FIGURE 7.

## DETAILED DESCRIPTION

**020** Throughout this description, embodiments and variations are described for the purpose of illustrating uses and implementations of the invention. The illustrative description should be understood as presenting examples of the invention, rather than as limiting the scope of the invention.

**021** FIGURE 1 is a block diagram of a cellular system segment 10 in which the present invention may be employed. In this cellular system segment 10, there are a plurality of base stations 22, 24, 26, 42, and 44 that are geographically separated. The cellular system segment 10 also includes a mobile unit 30. The mobile unit may be any wireless apparatus, including a cellular Modulator/Demodulator ("modem") that may communicate with a base station (BS) 22, 24, 26, 42, or 46. For example, the mobile unit may be a cellular telephone, personal data assistant (PDA), or computer. In one embodiment, each base station may communicate with the mobile unit when the signal strength of the mobile unit, as received at a base station, is sufficiently strong. In one embodiment, the base stations support the same wireless communication protocol standard (such as a third generation (3G) Code Division Multiple Access (CDMA) standard). In another embodiment of the invention, the base stations may support different or multiple communication protocol standards. In addition, the mobile unit 30 may support a single or multiple communication protocol standards. For example, the device 30 may support a CDMA standard, Advanced Mobile Phone Service (AMPS) standard, a Time Division Multiple Access (TDMA) standard, and a Groupe Spécial Mobile (GSM) standard. In the example shown in FIGURE 1, the mobile unit 30 is capable of communicating with all of the base stations 22, 24, 26, 42, or 44 using a 3G CDMA standard.

**022** In FIGURE 1, after acquiring a pilot signal from a base station 22, 24, and 26, the mobile unit 30 may communicate with the base station. Under the 3G CDMA standard, a base station may produce an interleaved, FEC encoded signal and transmit this signal to the mobile unit 30. The mobile unit 30 would de-interleave and decode the signal from the 3G CDMA standard compliant base station.

**023** FIGURE 2 illustrates a block diagram of an exemplary base station 100 that may be used in practicing the present invention. The exemplary base station 100 may include a CPU 102, a RAM 104, a ROM 106, a storage unit 108, an antenna 113, a first modem/transceiver 112 and a second modem/transceiver 114. The first modem/transceiver

112 may couple, in a well-known manner, the base station 100 to a central cellular network control center via an Internet connection or via a wired telephone system such as the Plain Old Telephone System ("POTS"). The second modem/transceiver 114 may couple the base station 100 to the cellular network segment 10. The modem/transceiver 114 may be an Ethernet modem, telephone modem, wireless modem or other communication device that communicates with the cellular network 10 (FIGURE 1) via the antenna 113. The CPU 102 directs communications between the first and second modem, 112 and 114, respectively, for messages between the central network control center, Internet, or POTS, and one or more mobile units.

024    The ROM 106 may store program instructions to be executed by the CPU 102. The RAM 104 may be used to store temporary program information and overhead information for other base stations in its sector (*i.e.,* nearby base stations). The storage unit 108 may comprise any convenient form of data storage and may be used to store the overhead information. An exemplary portion of the modem/transceiver 114 is shown in FIGURE 4. FIGURE 4 is a simplified block diagram of an exemplary base station error encoder and interleaver apparatus 70 that may be used to practice the present invention.

025    In the 3G CDMA standard, the FEC encoder and interleaver form a turbo coder/decoder ("codec") that is employed by base stations and mobile units. The exemplary apparatus 70 is a turbo codec that includes a temporal interleaver 72, a first constituent encoder 74, and a second constituent encoder 76. In the apparatus 70, the FEC is comprised of two constituent systematic codes with a 1/3 coding rate whereby four parity check bits are generated for each information bit. The temporal interleaver 72 is employed before the second constituent encoder 76. The temporal interleaver 72 may be a turbo interleaver in an exemplary embodiment. In other embodiments, the apparatus 70 may have more than two constituent encoders and more than one temporal interleaver. In addition, other configurations, such as serial concatenation of these components 72, 74, and 76 may be employed in another embodiment of the invention.

026    FIGURE 3 is a block diagram of an exemplary mobile unit 120 that may be used to practice the present invention, in particular, that may be used to de-interleave and decode received FEC encoded, interleaved signals. The exemplary device 120 may include a central processing unit (CPU) 122, a random access memory (RAM) 124, a read only memory

(ROM) 126, a display 128, a user-input device 132, a transceiver application specific integrated circuit (ASIC) 134, a microphone 138, a speaker 136, and an antenna 133. The ROM 126 is coupled to the CPU 122 and stores the program instructions executed by the CPU 122. The RAM 124 is coupled to the CPU 122 and stores temporary program data and overhead information. The user-input device 132 may comprise any convenient input device such as a keypad, touch pad screen, track ball or other similar input device that allows the user to navigate through menus in order to place calls, in addition to performing other functions. The display 128 is an output device such as a CRT, LCD or other similar screen display that enables the user to read received data and to place calls.

**027** The microphone 138 and speaker 136 may be incorporated into a handset that is coupled to the ASIC 134. The microphone 138 and speaker 136 may also be separated from the handset to allow hands-free communication. In this mode, the ASIC 134 may include voice activation circuitry that converts voice commands into data. The data is transmitted to the CPU 122 via a serial bus 139 and may include a telephone number to be dialed.

**028** The transceiver ASIC 134 includes an instruction set necessary to communicate data and voice signals over the cellular network segment 10 (FIGURE 1). In one embodiment, the transceiver ASIC 134 is a 3G code-division multiple-access (CDMA) ASIC and the cellular network is a CDMA network that supports data and voice communication. The ASIC 134 is coupled to the antenna 133 to communicate signals within the cellular network segment 10. When a data signal is received by the transceiver ASIC 134, the data is transferred to the CPU 122 via the serial bus 139. The ASIC 134 may perform operations to de-interleave and decode received interleaved, FEC-encoded signals. An exemplary portion of the ASIC 134 is shown in FIGURE 5A.

**029** As shown in FIGURE 5A, the exemplary ASIC 134 includes a radio frequency (RF) circuit 62, a Rake receiver 64, a searcher correlator 66, a P/N sequence generator 51, a demodulator processor 55, and a deinterleaver & Turbo decoder 53. These components, 51, 53, 55, 62, 64, and 66 for CDMA mobile units are well known to those of skill in the wireless communications arts.

**030** FIGURE 5B is a block diagram of a mobile unit deinterleaver and error decoder apparatus 80 that may be employed in the transceiver ASIC shown in FIGURE 5A for a CDMA based system. The mobile unit deinterleaver and error decoder apparatus 80 includes a

channel deinterleaver 82, a first and a second A Posteriori Probability ("APP") decoder 84 and 96, respectively, an interleaver buffer 86, a deinterleaver buffer 88, an adder 92, and a hard decision unit 94. The channel deinterleaver 82 stores received FEC-encoded, interleaved data. The first and second decoders 84 and 96, respectively, operate recursively by feeding the output of the second decoder 96 to the input of the first decoder 84. The apparatus 80 is thus an iterative decoder. The interleaver buffer 86 stores the data generated by the first APP decoder 84. The deinterleaver buffer 88 stores the data generated by the second APP decoder 96.

**031** The memory size used by the buffer varies as a function of the channel rate and encoding rate used. For example, for a CDMA IS-2000 standard based supplemental channel operating at 153.6kbps, encoded at ¼ rate, where each APP decoder output comprises 8 bits, the total buffer (or memory) size of 86 and 88 comprises 6Kbytes (*i.e.,* 2(buffers)x153.6(kbps)x20(ms)x8(bits)=6Kbytes). In the apparatus 80, the interleaver buffer 86 and the deinterleaver buffer 88 represent the major memory requirement of the apparatus. Turbo codes are usually implemented for high data rate, which translates to a large memory requirement. Therefore, proper design of the interleaver and deinterleaver in a turbo decoder is crucial to the size of the modem IC. Also, because of the iterative decoding needed for decoding turbo code, economic use of the decoding engine is also an important design issue. It is noted that memory power consumption levels vary as a function of the memory structure. In particular, a single read/write port memory structure is generally more power and IC real estate efficient than is a dual read/write port memory structure.

**032** FIGURE 6A is a block diagram of an exemplary mobile unit deinterleaver and error decoder apparatus 140 that may be employed in the transceiver ASIC shown in FIGURE 5A. Similar to apparatus 80, apparatus 140 may be employed to decode and de-interleave received signals encoded with a 1/3 coding rate constituent codes. However, apparatus 140 may be modified for any coding rate. As shown in FIGURE 6A, the apparatus 140 includes a channel deinterleaver buffer 142, an input buffer 144, a single APP decoder engine 146, a first multiplexer 148, a (de)interleaver memory 150, a second multiplexer 162, a sign coder 164, and an interleaver buffer address controller 166. The channel deinterleaver buffer 142 stores interleaved, FEC encoded data. The APP decoder engine 146 functions as the first and the second A Posteriori Probability ("APP") decoder 84 and 96 of FIGURE 5B.

**033**     In the apparatus 140, the single APP decoder engine 146 and one information data estimation frame size buffer or memory 150 are used to decode a constituent code 1 and a constituent code 2. The channel deinterleaver buffer 142 stores all of the received turbo coded (FEC-encoded) symbols that are corrupted by the channel. The channel buffer 142 data is combined with data that is read from the (de)-interleaver buffer or memory 150 to form input to the APP decoder 146 via the input buffer 144. The decoder engine 146 calculates the *a posteriori* probability of each information data in a manner that is known to one of skill in the art. In particular, for each FEC or turbo error decoding iteration, the APP decoder engine 146 first decodes constituent code 1. Data deinterleaving prior to decoding is achieved by reading data from the (de)-interleaver buffer for use by the APP decoder engine 146 and writing constituent code 2 data to the (de)-interleaver buffer in the last iteration. The APP decoder engine 146 then decodes constituent code 2. Data interleaving is achieved by the combination of reading data from the (de)-interleaving buffer or memory 150 for use by the APP decoder engine 146 and then writing data generated by the APP decoder engine 146 to the (de)-interleaving buffer or memory 150 while decoding constituent code 1 in the last iteration.

**034**     FIGURE 7 depicts an exemplary process 170 that the apparatus 140 may perform when de-interleaving and decoding a received interleaved, error encoded signal in accordance with the teachings of the present invention. It is noted that the (de)-interleaving buffer or memory contains a single identical value, *e.g.*, every address is cleared to 0 prior to the start of process 170. In step 171, data is read from the memory 150 by using a read address that is generated row-by-row with column shuffling, hereafter referred as a "horizontal addressing" algorithm. The data stream produced at step 171 is thereby effectively de-interleaved, *i.e.* in the original pre-interleaved order. Then the data is delayed by a fixed number of clock cycles D1 (step 172), where D1 is determined by the interleaving dimension and the decoding delay. The process 170 then retrieves data from the channel deinterleaver buffer 142 (step 173) so that retrieved channel deinterleaver buffer data, representing received coded symbols, is synchronized with the delayed data from the (de)-interleaving buffer/memory 150 (FIGURE 6A). The first constituent code, Recursive Systematic Convolutional 1 ("RSC1") is decoded (step 174) by processing data from the memory 150 (FIGURE 6A) and from the channel deinterleaver buffer 142 (FIGURE 6A) according to a constituent code 1 trellis. The decoded RSC1 data is written into the memory 150 on a row-by-row basis, with the column shuffling

algorithm (step 175). Steps 171 through 175 are repeated until a complete frame has been processed (step 176).

**035** Data is then read from the memory 150 using a read address that is generated on a column-by-column basis using a row shuffling algorithm (step 177). The data stream that is produced at this step (177) is effectively interleaved. The data is then delayed by a fixed number of clock cycles, D2 (step 178), where D2 is determined by the interleaving dimension and the decoding delay. D2 is generally not the same as D1 although in some embodiments it can be. The process 170 then retrieves data from the channel deinterleaver buffer 142 (step 179) so that retrieved channel deinterleaver buffer data, representing received coded symbols, is synchronized with the delayed data from the (de)-interleaving buffer/memory 150. The second constituent code (RSC2) is then decoded (step 180) by processing data received from the memory 150 and from the channel deinterleaver buffer 142 according to a constituent code 2 trellis. The decoded RSC2 data is written into memory 150 on a column-by-column basis, with the row shuffling algorithm (step 181). Steps 177 through 181, inclusive, are repeated until a complete frame has been processed (step 182). Steps 171 through 182 are then repeated until the last decoding iteration (step 183), in an exemplary embodiment, until the decoding iteration reaches a predetermined threshold number. A hard decision is then made between the computed data and the data stored in the memory 150 (step 184). Note that steps 171-175 (and steps 177-181) are only serially performed for a specific data symbol. During operation in a data symbol pipeline the steps 171-175 (and steps 177-181) may be performed contemporaneously.

**036** The (de)interleaver memory 150 (FIGURE 6A) includes four, equally sized sub-buffers or memories 152, 154, 156, and 158. In the exemplary embodiment, the total size of the memory 150 is equal to one information data estimation frame. During a decoding operation, the four sub-buffers 152, 154, 156, and 158 store information data estimations from a current decoding iteration or from a previous decoding iteration. In the apparatus 140, the APP decoder engine 146 operates continuously on the data in these buffers, *i.e.*, without waiting for data to be interleaved or deinterleaved as in the apparatus 80. Data interleaving and de-interleaving is performed when information data estimation is saved and read from the memory 150 via addresses that are generated by the interleaver buffer address control unit 166. In the exemplary apparatus 140, the memory size is 3 KB (*i.e.*,

153.6(kbps)x20(ms)x8(bits), assuming that 8 bits represent each soft value of information data estimation. Accordingly, the apparatus 140 requires the half of the memory required by the apparatus 80 for information data estimation, *i.e.,* 3 Kbytes versus 6 Kbytes.

**037** It is noted that apparatus 140 achieves the memory size reduction regardless of whether the memory 150 is subdivided into four buffers. The subdivision of the Memory 150 in the exemplary apparatus 140, however, enables use of single read/write memory port types, which reduces power consumption and silicon real estate (as compared to dual memory port types). In particular, during operation of the apparatus 140, at any given time instance, only one sub-buffer is being read and only one sub-buffer is being written while the two other sub-buffers are disabled. The apparatus 140 may be used to implement any FEC or turbo coder that utilizes a matrix interleaver algorithm. In particular, the entire data frame that is to be interleaved is virtually arranged as a (row_number) x (column_number) buffer in memory 150, where "column_number" is a design parameter, "row_number" = ceiling( Frame_length / column_number), and "Frame_length" is the number of bits in  the information data estimation frame.

**038** FIGURE 6B depicts an exemplary interleaving process 200 that may be used to interleave data when the data is stored in the memory 150 in accordance with the teachings of the present invention. In the exemplary interleaving process 200, data is written into the memory matrix on a row-by-row basis (step 202). The content of each row is then permutated using a specific pattern, column shuffling in a CDMA IS-2000 based system (step 204). Then every row, as a whole, is permutated using another specific pattern, row shuffling in the CDMA IS-2000 based system (step 206). To complete the interleaving process 200, data is read from the memory matrix 150 on a column-by-column basis (step 208).

**039** FIGURE 6C depicts an exemplary de-interleaving process 210 that may be used to deinterleave data when the date is stored in the memory 150 in accordance with the teachings of the present invention. In the exemplary de-interleaving process 210, data is written into the memory matrix on a column-by-column basis (step 212). Every row, as a whole, is then permutated using another specific pattern, reverse row shuffling in the CDMA IS-2000 based system (step 214). The content of each row is then permutated using a specific pattern, reverse column shuffling in a CDMA IS-2000 based system (step 216). To complete the de-interleaving process 210, data is read from the memory matrix 150 on a row-by-row basis

(step 218). It is noted that any specific column and row shuffling algorithm may be used in the present invention as a function of how the received data is/was interleaved.

**040**     In an exemplary embodiment, the interleaver buffer address controller 166 (shown in FIGURE 6A) generates the read and write addresses used for turbo interleaving and deinterleaving. In one embodiment, the controller 166 generates addresses using two methods (processes 200 or 210 as described above with reference to FIGURES 6B and 6C) as a function of which constituent code the APP decoder is processing. In particular, when the APP decoder engine 146 (FIGURE 6A) is decoding constituent code 1, the address controller 166 generates read and write addresses on a row-by-row basis, with column shuffling, as required by de-interleaving read and interleaving write (process 200 shown in FIGURE 6B). When the APP decoder engine 146 is decoding constituent code 2, the address controller 166 generates read and write addresses on a column-by-column basis, with row shuffling, as required by interleaving read and de-interleaving write (process 210 shown in FIGURE 6C). Alternatively, employing these methods (200 and 210) enables the address controller 166 to interleave information data estimations between the APP decoder engine code 1 output and the APP decoder engine code 2 input. Using these methods also allows the address controller 166 to deinterleave information data estimations between the APP decoder engine code 2 output and APP decoder engine 146 code 1 input.

**041**     In an exemplary embodiment, the single port memory may be used for the frame buffer or memory 150. In particular, in an exemplary embodiment, the memory 150 includes four, quarter frame size single port memories (152, 154, 156, and 158), In this embodiment, these memories may be employed provided that a specific delay between the buffer read and write cycles is selected and a specific process to generate read and write addresses is employed.

**042**     FIGURE 8 is a diagram of an exemplary memory mapping/layout 190 for use by the apparatus shown in FIGURE 6A and the process depicted in FIGURE 7. In this memory layout, the memory 150 is divided into four, quarter frame sub-buffers. In FIGURE 8, the sub-buffers 00, 01, 10, and 11 are represented by four patterns. As shown in FIGURE 8, the memory 150 is divided into N x K segments or units where there are L groups of N rows and M groups of K columns or L x M segments. In the exemplary memory layout 190, each sub-buffer is formed from non-adjacent row and non-adjacent column units or segments form one

sub-buffer, resulting in 4 sub-buffers. Accordingly, each sub-buffer is physically separate from the other memories and may have a single address and a single port for reading or writing of data.

**043** In an exemplary embodiment, the specific delay between the read and write cycles enables data reading and writing in two different, advantageously adjacent row or column groups. In particular, when horizontal (row-by-row) addresses are being generated, the data writing address is ideally N rows greater than the data reading address. In this addressing scheme, one of the sub-buffers 00 or 01 and one of sub-buffers 10 or 11, are either read or written. The other two sub-buffers are then idle. When vertical (column-by-column) addresses are being generated, the data writing address is K columns greater than the data reading address. In this addressing scheme, one of sub-buffers, 00 or 10, and one of sub-buffers 01 or 11, are either read or written. The other two sub-buffers are then idle.

**044** The application of the memory layout 190 to the address controller 166 of FIGURE 6A requires appropriate write and enable signal generation (wo, E0, w1, E1, w2, E2, w3, and E3) and read and write address generation ("raddr" and "waddr", respectively). For example for vertical addressing of sub-buffer 00 and sub-buffer 01 (when sub-buffer 00 (index 0) is read, sub-buffer 01(index 1) is written, and sub-buffers 10 (index 2) and 11 (index 3) are idle, the signals generated by the address controller 166 may include: $w0 = 1$, $w1 = 0$, $w2 = 0$, $w3 = 0$; $E0 = 1$, $E1 = 1$, $E2 = 0$, $E3 = 0$;, Add0 = waddr (write address); Add1 = raddr (read address); Add2 = don't care; Add3 = don't care; and RS (read selection) = 1.

**045** The selection of row, N and column, K of the sub-buffer is then based on the interleaving matrix dimension and decoding delay. In detail, a matrix dimension that may be modeled in the present invention may have $(2^{(n1+n2)})$ rows and $(2^{(m1+m2)})$ columns; where n1, n2, m1, and m2 are integer numbers greater than 0. In this example, the address for interleaver matrix is (n1+n2+m1+m2) bits in length, while the address for corresponding sub-buffers is accordingly ((n1+n2+m1+m2) -2) bits in length by properly extracting two address bits for sub-buffer selection. For example, when N and K are both odd numbers, the following method can be used. For an interleaver matrix address denoted as A1(n1+n2+m1+m2), .....A(1), A(0), indexed row-by-row, from the first position in the first row, to the last position in the last row, the sub-buffer address is correspondingly obtained by removing A(n1+n2) and

A(0). These two address bits form the binary index of the sub-buffer that is selected, *i.e.* when A(n1+n2), A(0) = 1,0 sub-buffer 10 (index 2) is selected.

**046**    Different memory layouts may be selected as a function of the interleaving and FEC encoding process applied to the transmitted (received at the mobile unit) signal. For example, the above invention is extendable to other concatenated coding schemes including parallel concatenation and serial concatenation. The invention is also extendable to more than two component codes in the FEC-encoding scheme, and to more than one interleaving scheme. The same single APP decoder engine and single buffer architecture may be employed in these cases with appropriate modifications to control flow. In particular, the control flow may be expanded from two decoding stages to T stages, where T is equal to the number of constituent codes and the row shuffling and column shuffling algorithms may be modified to match interleaver algorithms employed at corresponding stages of encoding during the received signal generation (at a base station in one embodiment).

**047**    Thus, the previous description of the preferred embodiments is provided to enable any person skilled in the wireless communications art to make or use the present invention. The various modifications to these embodiments will be readily apparent to those skilled in the art, and the generic principles defined herein may be applied to other embodiments without the use of inventive faculty. Thus, the present invention is not intended to be limited to the embodiments shown herein, but is to be accorded the widest scope consistent with the principles and novel features disclosed herein.

**048**    While this invention has been described in terms of a best mode for achieving the objectives of the invention, it will be appreciated by those skilled in the wireless communications art that variations may be accomplished in view of these teachings without deviating from the spirit or scope of the present invention. For example, the present invention may be implemented using any combination of computer programming software, firmware or hardware. As a preparatory step to practicing the invention or constructing an apparatus according to the invention, the computer programming code (whether software or firmware) according to the invention will typically be stored in one or more machine readable storage mediums such as fixed (hard) drives, diskettes, optical disks, magnetic tape, semiconductor memories such as ROMs, PROMs, *etc.*, thereby making an article of manufacture in accordance with the invention. The article of manufacture containing the computer

programming code is used by either executing the code directly from the storage device, by copying the code from the storage device into another storage device such as a hard disk, RAM, *etc.*, or by transmitting the code on a network for remote execution.